An integrated System Development Approach for Mobile Machinery in consistence with Functional Safety Requirements

Dipl.-Ing. Erik Lautner

HYDAC System GmbH, MSC Entwicklungsbüro Berlin, Zum Kiesberg 16, 14979 Großbeeren (bei Berlin), Germany, E-mail: erik.lautner@hydac.com

Dipl.-Ing. Daniel Körner

HYDAC System GmbH, MSC Entwicklungsbüro Berlin, Zum Kiesberg 16, 14979 Großbeeren (bei Berlin), Germany, E-mail: daniel.koerner@hydac.com

Abstract

The article identifies the challenges during the system and specifically the software development process for safety critical electro-hydraulic control systems by using the example of the hydrostatic driveline with a four speed transmission of a feeder mixer.

An optimized development approach for mobile machinery has to fulfill all the requirements according to the Machinery Directive 2006/42/EC, considering functional safety, documentation and testing requirements from the beginning and throughout the entire machine life cycle.

The functionality of the drive line control could be verified in advance of the availability of a prototype by using a “software-in-the-loop” development approach, based on a MATLAB/SIMULINK model of the drive line in connection with the embedded software.

KEYWORDS: Functional Safety, Safety Related Development, V-Model, Modular Software Design, Testability, Software Test, Software-in-the-loop

1. Introduction

A contemporary development process for mobile machinery control systems has to include all necessary steps to comply with the increased safety requirements according to the Machinery Directive 2006/42/EC **/1/**. To be able to fulfill state of the art control systems safety requirements, it is necessary to consider the safety requirements right from the beginning and to look at the system with its different elements in its entirety.

Besides the big challenges fulfilling the safety requirements in view of the electro-hydraulic system architecture, the development of the embedded control software is exceptionally challenging. A specific focus and a high effort during the development

process are particularly related to the extensively enhanced documentation and testing requirements, which are often not directly linked to the development process.

The development process of safety critical systems and specifically of embedded software for mobile machinery has become a high complexity under consideration of the Machinery Directive 2006/42/EC **/1/** requirements. For this reason an optimized development approach for mobile machinery market has to fulfill all of the requirements according to the Machinery Directive 2006/42/EC **/1/** considering: control system design, software development, documentation and testing requirements throughout the entire machine life cycle. A furthermore growing standardization of subsystems and software modules could gain additional savings within the development process.

Using the example of a hydrostatic driveline with a four speed transmission of a feeder mixer (see **Figure 1**) the article would like to point out the complex challenges and specifics of the safety related software development process.

Figure 1: Feeder mixer

The particular drive line project was realized using the standard “Safety of machinery - safety-related parts of control systems” EN ISO 13849-1:2008 **/2/** as a basis. A maximum required performance level PL_r ‚d‘ was demanded by the safety functions of hydrostatic drive line including a four speed power shift transmission.

2. Safety Related Development Process

The development process of a safety critical control system includes e.g.:

- identification of functional and safety requirements
- technical safety concept and system definition
- control software development
- software and system verification, validation and documentation

The safety related development process, which has to be considered in the project, is based on a simplified V-Model according to EN ISO 13849 **/2/** and BGIA-Report 2/2008 **/6/.** The applied V-Model is shown in **Figure 2**.

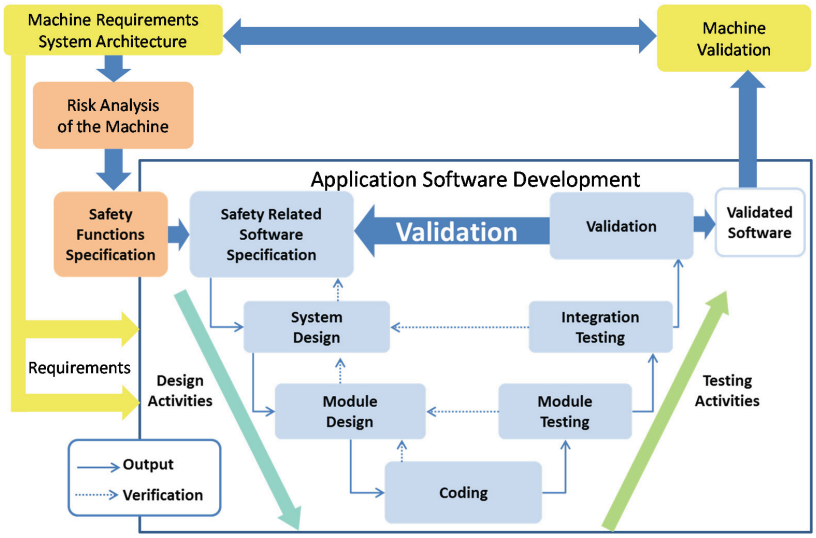

Figure 2: Safety related development process with simplified V-Model acc. to EN ISO 13849 **/2/** and BGIA-Report 2/2008 **/6/**

The safety related development process requires extensive new requirements regarding test, documentation and traceability. It is of key importance to consider the demands of testability and documentation from the very beginning of the development process. For an efficient work process it is reasonable to establish an integrated development environment, which is generally capable to fulfill all of these demands.

It is strongly recommended by the functional safety standards, to support the development activities for safety related systems by an appropriated tool chain, in order to realize an increased software quality, efficient error avoidance, a good reproducibility and an established traceability during the software development.

3. MATCH - “Mobile Application Tool Chain”

In response to the today´s challenges, HYDAC has developed an integrated tool chain intended as a comprehensive approach to system and software development of mobile machines. The **M**obile **A**pplication **T**ool **C**hain, which is named MATCH, is an integrated software development environment with a specific focus on mobile machinery.

The MATCH tool chain consists on the PC level of three different development tools: Project Definition Tool (PDT), Machine Service Tool (MST) and Test and Simulation Environment (TSE). The individual tools have been precisely synchronized to allow loss free transition of information. Interface losses and multiple manual entries of information are almost completely avoided. It addresses all parts of the V-model development process starting with the overall vehicle safety and functional requirements, including documentation, software testing, commissioning and optimization on the vehicle as well as diagnostics and field service (see **Figure 3**).

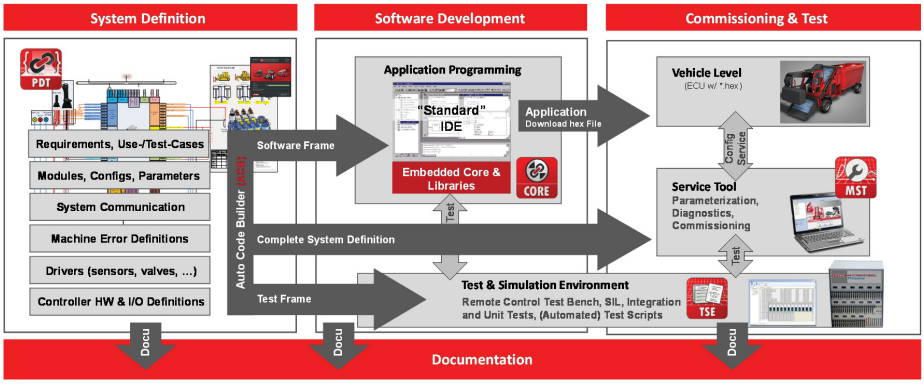

Figure 3: Integrative tool chain approach with MATCH

Based on the system definition the tool chain provides embedded software functionality by code generation using an Auto Code Builder (ACB). The automatically generated code applies to a certified embedded core software (Embedded Core) and library blocks (Libraries) specifically tailored to mobile machinery requirements.

The embedded core software supports e.g. the needs regarding communication, NvMem parameter management, failure management, diagnostic functionality and software modularization. The hardware adaption layer supports besides the specific BSP needs moreover the safety system of the used embedded hardware platform. **Figure 4** shows, that the output of the auto code generation is an embedded software frame per each controller of a complete vehicle project with more than one controller. The specific vehicle application functionality can be easily implemented into this frame.

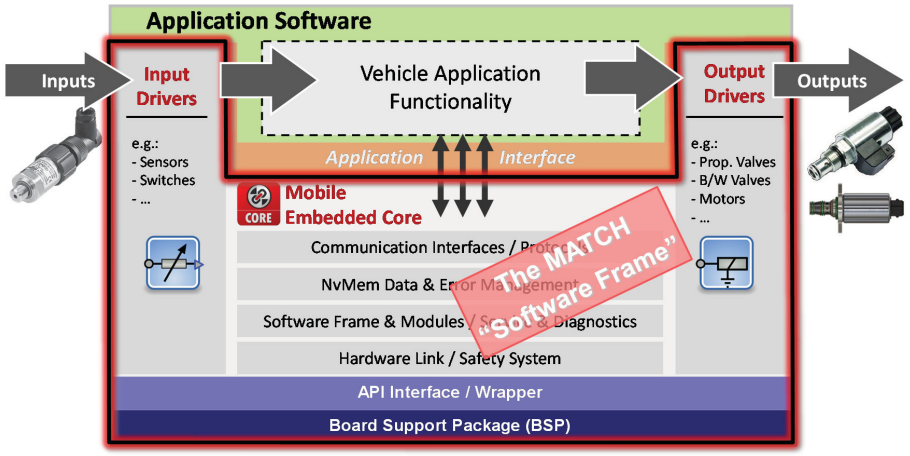

Figure 4: MATCH auto-generated embedded software frame

The incorporated embedded software development process is hardware-independent and provides multi-controller capability. The code generation provides fully tested and documented robust more than code. By using the described approach the programming effort for the specific application can be reduced drastically.

The safety related elements of the MATCH software development environment incorporate a certification according the following standards: IEC/EN 61508:2010 (SIL 2), EN ISO 13849:2008 (PL ‚d'), ISO 25119:2010 (AgPL ‚d') and EN 16590:2014 (AgPL ‚d').

4. Drive Line Functionality & System Layout

The new tool chain has been set to work in several system development projects. As an example this paper highlights the electro-hydraulic control system development of the hydrostatic driveline with a four speed transmission of a feeder mixer.

The driveline of the feeder mixer attains a maximum speed of 42 kph on public roads. It supports a vast range of functional features in addition to different driving modes. These include e.g.: cruise control, speed limitation, over speed protection, anti-stall, power limitation, hill starting and auto-park brake. The electro-hydraulic layout of the drive line is shows **Figure 5**.

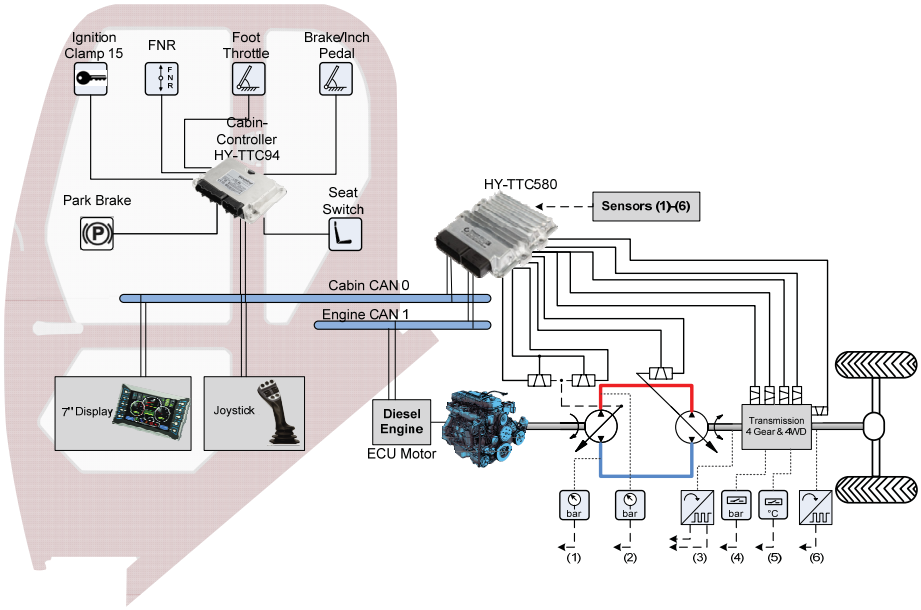

Figure 5: Electro-hydraulic system layout of the drive line

It is particularly significant to consider the safety requirements of a control system from the very beginning of the design process. In order to be able to identify the safety functions of the drive line functionality, it was of primary necessity to apply a hazard & risk analyses. The H&R analysis were realized acc. to the standard EN ISO 13849 **/2/**. It was identified a maximum required performance level of PL_r ‚d' (see **Table 1**).

#	Malfunction	PLr
Drive line		
HD1	Undesired starting at engine start	d
HD2	Unexpected start while engine running	d
HD3	Start in unexpected direction	c
….	…	…
Cruise control		
HC1	Unexpected ground speed value	c
HC2	Impossible deceleration	c
…	…	

Table 1: Exemplary hazard & risks

Table 2 shows exemplary some of the related safety functions of different types: "Safe Torque Off" (STO), "Safe Direction" (SDI) and "Safe Speed Range" (SSR). The specific safety functions will be exemplary used to investigate more detailed the software test requirements.

Safety Function	Description	Type	PL_r	Reaction Time
Drive line				
SF_D-01	Prevent unintended start of movement	STO	d	500 ms
SF_D-02	Safe demand of driving direction	SDI	c	1000 ms
…	…	…	…	…
Cruise Control				
SF_C-01	Safe speed presetting	SSR	c	1000 ms
…	…	…	…	…

Table 2: Safety Functions

5. Software Architecture of the Safety Functions

The software development was done acc. to the simplified V-model requirements considering a maximum required performance level of PL_r ‚d'. Using the MATCH tool chain enabled us to focus the capacities specifically on the application software development. The frame functionality, generated by an auto code builder, is already certified. The exemplary design of the drive line control function is shown in **Figure 6**.

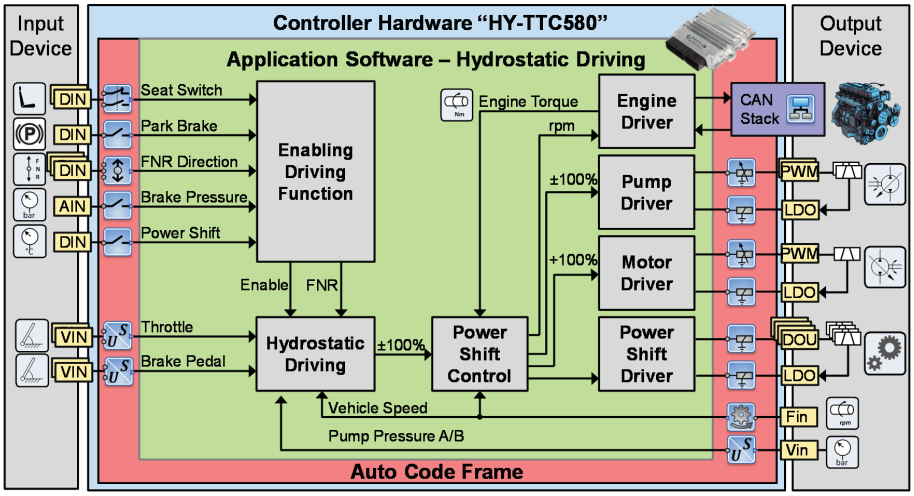

Figure 6: Major software modules of the drive line function

Due to the fact that safety requirements for software are entirely process related, the testability and documentation of safety related software is of key importance. In order to achieve testable software, test features have to be implemented into the software by designing testable functional modules, limiting the logic combinations between functions, implementing test interfaces and entry points into the code. Therefore the testability of the software has to be defined at the very beginning of the software development process.

6. Test Methods, Test Automation & Testability

It is shown by the V-model, that the testing activities are a stepwise procedure, which addresses directly the different steps of design activities on the left side of the V-model. The test methods can be differentiated into:

- unit / module tests
- integration tests
- system tests
- validation tests

By using the so called “bottom-up” principle, the test activities are usually starting with the smallest software component, a simple software function. For this reason the software unit / module test is the first step within the complete testing process.

Figure 7 shows the test concept for the unit, module and integration tests based on the used MATCH tool chain. The automated test scripts can be defined in the programming language “Python”. The important advantage of the Test & Simulation Environment (TSE) is, that based on the automatic code generation, automatically all of the software frame definitions are as well in the testing environment available: PIN´s, CAN messages, error codes, parameters, defines, enumerations (see **/7/**).

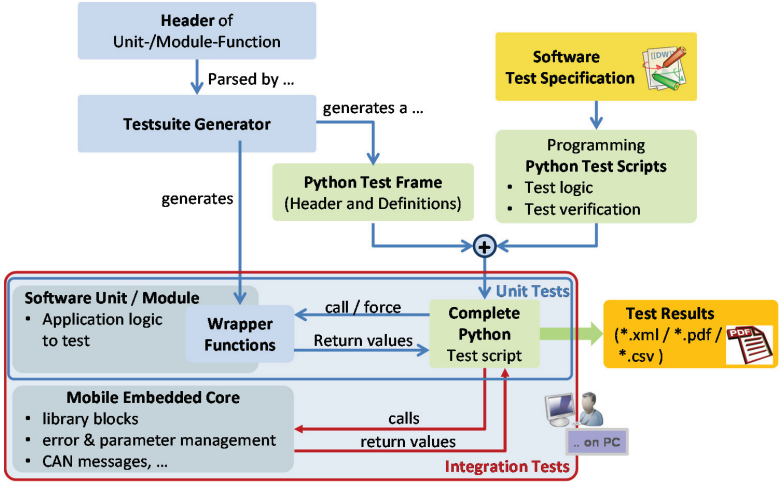

Figure 7: Test concept for unit, module and integration tests **/9/**

Figure 7 shows, that for an efficient unit and module testing, the test frame should be generated automatically into the Python test environment exclusively based on the available C-function header information of the application project.

If there is a specific need to handle machine options and configurations, a test environment has to be capable to allow:

- a reconfiguration of the ECU PIN´s,
- a restart of the software (ignition on/off),
- an emulation of the NvMem within the PC storage,
- an access to NvMem parameters (change configurations, characteristics) and
- an access to the error management (failure mode testing)

Figure 8 shows the approach for integration testing on PC. The application can be executed together with the MATCH embedded software frame. The hardware BSP will be replaced by a simulation API, which allows an access to the virtual PIN values, the board information and to the CAN communication as well.

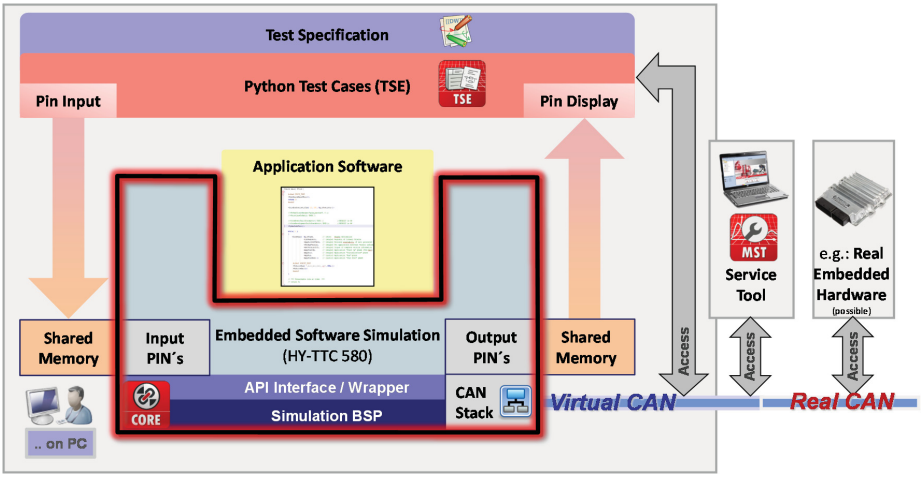

Figure 8: Integration test approach

The embedded software simulation is executed on PC and commanded by the Python test scripts. For this reason the Python environment is running in TSE environment **/7/** and has access to the application, the MATCH embedded frame, the CAN messages and to the diagnostic interface. The integration test environment is also capable to communicate via virtual CAN with external controller hardware.

7. Software-in-the-loop

The control functionality of the hydrostatic drive, together with the 4-speed shift transmission, could be verified by using a “software-in-the-loop” (SIL) development approach. This enabled the developer immediately to apply software testing methods to the embedded application software code in advance of the availability of a prototype.

The “software-in-the-loop” testing approach is moreover particularly suitable for the validation of very complex safety functions (e.g. cruise control “Safe speed presetting”), which are strongly influenced by the vehicle dynamics.

A MATLAB/SIMULINK model of the complete driveline in connection with the real embedded control software was used for the “software-in-the-loop” investigations with a direct interface to the tool chain. A simplified model of the drive line (w/o sensors and control loops) shows **Figure 9**.

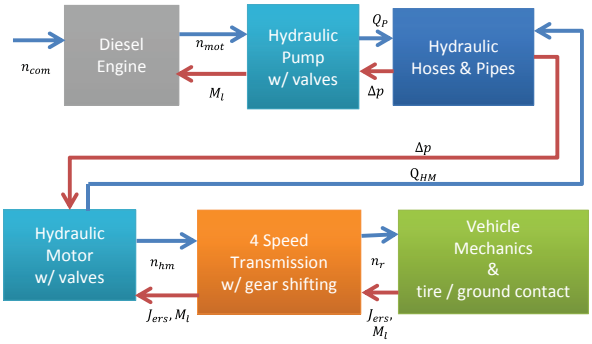

Figure 9: Drive line simulation model in MATLAB/SIMULINK (w/o sensors, …)

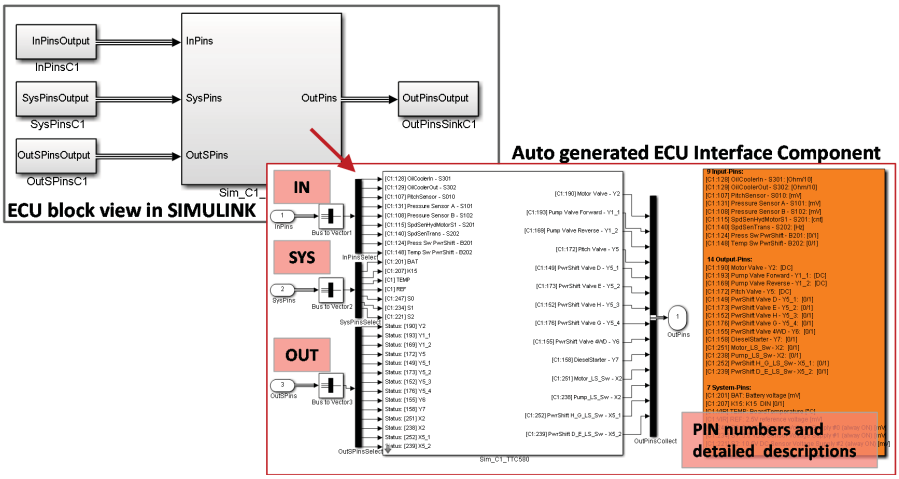

Figure 10: Auto generated ECU interface block in SIMULINK

The interface from SIMULINK to the embedded software is managed by a specific ECU and CAN message interface to the embedded simulation. The applied method is analog to the integration testing approach. SIMULINK blocks for the ECU-pin-access and the CAN message interface were automatically generated based on the output data of the Auto Code Builder (see **Figure 10**).

The complete “software-in-the-loop“-model in SIMULINK shows **Figure 11**.

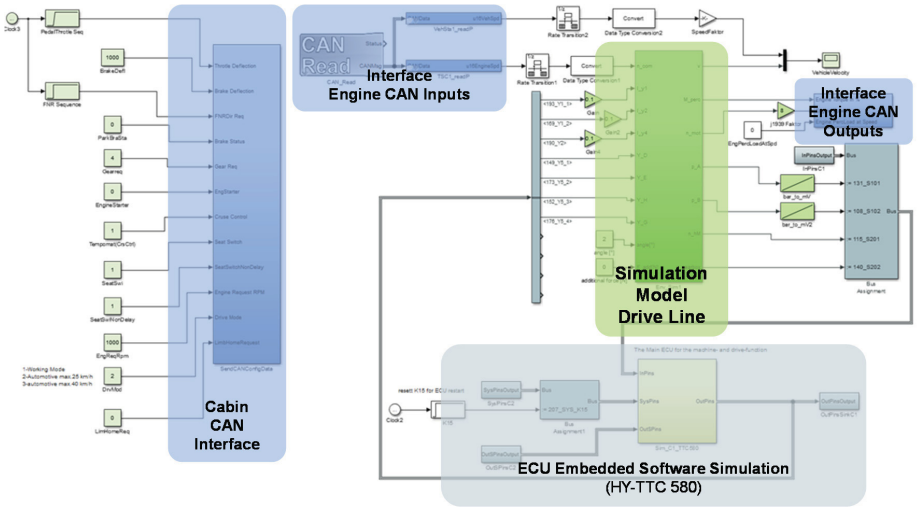

Figure 11: “Software-in-the-loop” model w/ an interface to the embedded software

8. Summary and Outlook

In view of the enhanced safety requirements driven by the Machinery Directive 2006/42/EC **/1/**, it has been shown that an integrated tool chain approach can reduce the development time noticeable by using a consistent project database in connection with a configurable middle ware, which is already certified to the different functional safety standards of mobile machinery and specifically adjusted to their needs.

Furthermore particularly the extensive documentation requirements and the complete range of testing methods should be an integrated component of the used development environment.

A robust software design concept based on a multi-layer approach in connection with an auto code builder and library block concept was applied for the application software development. A clever modularization of the software is a prerequisite for the testability of the embedded software and has to be defined at the very beginning of the development.

A “software-in-the-loop”-approach enables the developer - prior to the commissioning of the real prototype vehicle - immediately to apply software testing methods to the embedded application software code. Using “software-in-the-loop” test method generates specific advantages, when designing safety functions with a complex dynamic feedback from the system.

9. References

/1/ Official Journal of the European Union, L157/24,EN,9.6.2006 “Machinery directive” DIRECTIVE 2006/42/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 17 May 2006 on machinery, and amending Directive 95/16/EC (recast), 2006

/2/ EN ISO 13849, Safety of machinery – Safety-related parts of control systems, EN ISO 13849-1:2008-12 Part 1: General principles for design Safety of machinery, 2008
EN ISO 13849-2:2013-02 Part 2: Validation, 2013

/3/ IEC/EN 61508:2010 , Parts 1 to 7 - Functional safety of electrical/ electronic/programmable electronic safety-related systems, Geneva, International Electrotechnical Commission

/4/ ISO 25119:2010(E), Part 1 to 4: Tractors and machinery for agriculture and forestry - Safety-related parts of control systems

/5/ DIN EN 16590:2014, Part 1 to 4: Tractors and machinery for agriculture and forestry - Safety-related parts of control systems

/6/ BGIA-Report 2/2008 Functional safety of machine controls - Application of EN ISO 13849. BGIA (today: IFA) – Institute for Occupational Safety and Health of the German Social Accident Insurance. Sankt Augustin, 2008

/7/ MATCH - „Mobile Application Tool Chain”, Software development environment for mobile machinery, User Manual, HYDAC System GmbH, 2016

/8/ Weltzien, C., Lautner, E.: Modular software design of safety related systems for mobile machinery. 14th Scandinavian International Conference on Fluid Power, Tampere, 2015

/9/ Körner, D.: Entwicklung von Teststrategien für sicherheitsrelevante Anteile der Steuerungssoftware hydrostatischer Fahrantriebe in mobilen Arbeitsmaschinen. Diplom, TU Dresden, 2015